2.29 Numerical Fluid Mechanics
Spring 2015 – **Lecture 22**

REVIEW Lecture 21:

- Time-Marching Methods and ODEs – IVPs: End
  - Multistep/Multipoint Methods: Adams Methods
    - Additional points are at past time steps
  - Practical CFD Methods
  - Implicit Nonlinear systems
  - Deferred-correction Approach
- Complex Geometries
  - Different types of grids
  - Choice of variable arrangements:
- Grid Generation
  - Basic concepts and structured grids
    - Stretched grids
    - Algebraic methods (for stretched grids), Transfinite Interpolation

TODAY (Lecture 22): Grid Generation and Intro. to Finite Elements

- Grid Generation
  - Basic concepts and structured grids, cont'd
    - General coordinate transformation
    - Differential equation methods
    - Conformal mapping methods
  - Unstructured grid generation
    - Delaunay Triangulation
    - Advancing Front method
- Finite Element Methods
  - Introduction
  - Method of Weighted Residuals: Galerkin, Subdomain and Collocation
  - General Approach to Finite Elements:
    - Steps in setting-up and solving the discrete FE system
    - Galerkin Examples in 1D and 2D
  - Computational Galerkin Methods for PDE: general case
    - Variations of MWR: summary
    - Finite Elements and their basis functions on local coordinates (1D and 2D)

References and Reading Assignments Complex Geometries and Grid Generation

- Chapter 8 on "Complex Geometries" of "J. H. Ferziger and M. Peric, Computational Methods for Fluid Dynamics. Springer, NY, 3rd edition, 2002"
- Chapter 9 on "Grid Generation" of T. Cebeci, J. P. Shao, F. Kafyeke and E. Laurendeau, *Computational Fluid Dynamics for Engineers*. Springer, 2005.
- Chapter 13 on "Grid Generation" of Fletcher, Computational Techniques for Fluid Dynamics. Springer, 2003.
- Ref on Grid Generation only:
  - Thompson, J.F., Warsi Z.U.A. and C.W. Mastin, "Numerical Grid Generation, Foundations and Applications", North Holland, 1985

Grid Generation for Structured Grids: General Coordinate transformation

- For structured grids, mapping of coordinates from Cartesian domain to physical domain is defined by a transformation: $x_\mathrm{i} = x_\mathrm{i}(\xi_\mathrm{j}) \quad (i \,\&\, j = 1, 2, 3)$

$$J = \det\left(\frac{\partial x_i}{\partial \xi_j}\right) = \begin{vmatrix} \frac{\partial x_1}{\partial \xi_1} & \frac{\partial x_1}{\partial \xi_2} & \frac{\partial x_1}{\partial \xi_3} \\ \frac{\partial x_2}{\partial \xi_1} & \frac{\partial x_2}{\partial \xi_2} & \frac{\partial x_2}{\partial \xi_3} \\ \frac{\partial x_3}{\partial \xi_1} & \frac{\partial x_3}{\partial \xi_3} & \frac{\partial x_3}{\partial \xi_3} \end{vmatrix}$$

- All transformations are characterized by their Jacobian determinant J.
  - For Cartesian vector components, one only needs to transform derivatives. One has:

$$\frac{\partial \phi}{\partial x_i} = \frac{\partial \phi}{\partial \xi_j}\frac{\partial \xi_j}{\partial x_i} = \frac{\partial \phi}{\partial \xi_j}\frac{\beta^{ij}}{J}, \quad \text{where } \beta^{ij} \text{ represents the cofactor of } \frac{\partial x_i}{\partial \xi_j} \text{ (element } i, j \text{ of Jacobian matrix)}$$

  - In 2D, $x = x(\xi,\eta)$ and $\phi = \phi(\xi,\eta)$, this leads to:

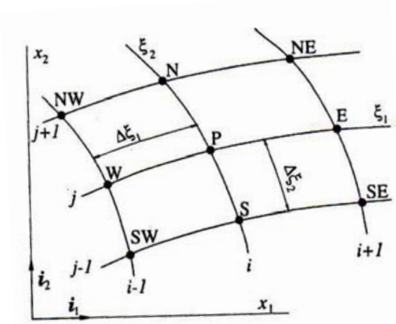
$$\frac{\partial \phi}{\partial x} = \frac{\partial \phi}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial \phi}{\partial \eta}\frac{\partial \eta}{\partial x} = \frac{\partial \phi}{\partial \xi}\frac{\beta^{11}}{J} + \frac{\partial \phi}{\partial \eta}\frac{\beta^{12}}{J} = \frac{1}{J}\left(\frac{\partial \phi}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial \phi}{\partial \eta}\frac{\partial y}{\partial \xi}\right)$$

Recall: the minor element m_{ij} corresponding to a_{ij} is the determinant of the submatrix that remains after the i^{th} row and the j^{th} column are deleted from $\mathbf{A}$. The cofactor c_{ij} of a_{ij} is: $c_{ij} = (-1)^{\mathrm{i+j}}\, m_{ij}$

Grid Generation for Structured Grids: General Coordinate transformation, Cont'd

- How do the conservation equations transform?
  The generic conservation equation in Cartesian coordinates:

$$\frac{\partial \rho\phi}{\partial t} + \nabla.(\rho\phi\vec{v}) = \nabla.(k\nabla\phi) + s_\phi \quad \Leftrightarrow \quad \frac{\partial \rho\phi}{\partial t} + \frac{\partial}{\partial x_j}\left(\rho\phi v_j - k\frac{\partial \phi}{\partial x_j}\right) = s_\phi$$



- becomes:

$$J\frac{\partial \rho\phi}{\partial t} + \frac{\partial}{\partial \xi_j}\left(\rho\phi U_j - \frac{k}{J}\left(\frac{\partial \phi}{\partial \xi_m}B^{mj}\right)\right) = J\, s_\phi$$

where:

$$U_j = v_k \beta^{kj} = v_1\beta^{1j} + v_2\beta^{2j} + v_3\beta^{3j} \quad \text{is proportional to the velocity component aligned with } \vec{\xi}_j \text{ (normal to } \xi_j = \text{const.)}$$

$$B^{mj} = \beta^{kj}\beta^{km} = \beta^{1j}\beta^{1m} + \beta^{2j}\beta^{2m} + \beta^{3j}\beta^{3m} \quad \text{are coefficients, sum of products of cofactors } \beta^{ij}$$

- As a result, each 1st derivative term is replaced by a sum of three terms which contains derivatives of the coordinates as coefficients
- Unusual features of conservation equations in non-orthogonal grids:
  – Mixed derivatives appear in the diffusive terms and metrics coefficients appear in the continuity eqn.

Structured Grids: Gen. Coord. transformation, Cont'd
Some Comments

- Coordinate transformation often presented only as a means of converting a complicated non-orthogonal grid into a simple, uniform Cartesian grid (the computational domain, whose grid-spacing is arbitrary)
- However, simplification is only apparent:
  - Yes, the computational grid is simpler than the original physical one
  - But, the information about the complexity in the computational domain is now in the metric coefficients of the transformed equations
    - i.e. discretization of computational domain is now simple, but the calculation of the Jacobian and other geometric information is not trivial (the difficulty is hidden in the metric coefficients)
- As mentioned earlier, FD method can in principle be applied to unstructured grids: specify a local shape function, differentiate and write FD equations. Has not yet been done.

Grid Generation for Structured Grids: Differential Equation Methods

- Grid transformation relations determined by a finite-difference solution of PDEs
  - For 2D problems, two elliptic (Poisson) PDEs are solved
  - Can be done for any coordinate systems, but here we will use Cartesian coordinates. The 2D transformation is then:
    - From the physical domain (x, y) to the computational domain (ξ, η)
    - At physical boundaries, one of ξ, η *is* constant, the other is monotonically varying
    - At interior points:

$$\frac{\partial^2 \xi}{\partial x^2} + \frac{\partial^2 \xi}{\partial y^2} = P(\xi,\eta)$$

$$\frac{\partial^2 \eta}{\partial x^2} + \frac{\partial^2 \eta}{\partial y^2} = Q(\xi,\eta)$$

  where $P(\xi,\eta)$ and $Q(\xi,\eta)$ are called the "control functions"

  - Their selection allows to concentrate the ξ, η lines in specific regions
  - If they are null, coordinates will tend to be equally spaced away from boundaries
  - Boundary conditions: ξ, η specified on boundaries of physical domain

Differential Equation Methods, Cont'd

- Computations to generate the grid mapping are actually carried out in the computational domain (ξ, η) itself !
  - don't want to solve the elliptic problem in the complex physical domain!
- Using the general rule, the elliptic problem is transformed into:

$$\alpha \frac{\partial^2 x}{\partial \xi^2} - 2\beta \frac{\partial^2 x}{\partial \xi \partial \eta} + \gamma \frac{\partial^2 x}{\partial \eta^2} + J^2 \left(P \frac{\partial x}{\partial \xi} + Q \frac{\partial x}{\partial \eta} \right) = 0$$

$$\alpha \frac{\partial^2 y}{\partial \xi^2} - 2\beta \frac{\partial^2 y}{\partial \xi \partial \eta} + \gamma \frac{\partial^2 y}{\partial \eta^2} + J^2 \left(P \frac{\partial y}{\partial \xi} + Q \frac{\partial y}{\partial \eta} \right) = 0$$

where $\alpha = x_\eta^2 + y_\eta^2\,; \quad \beta = x_\xi x_\eta + y_\xi y_\eta\,; \quad \gamma = x_\xi^2 + y_\xi^2\,; \quad J = x_\xi y_\eta - x_\eta y_\xi \quad (\text{with } x_\xi = \frac{\partial x}{\partial \xi}, \text{ etc})$

  - Boundary conditions are now the transformed values of the BCs in (x, y) domain: they are the values of the positions (x, y) of the grid points on the physical domain mapped to their locations in the computational domain
  - Equations can be solved by FD method to determine values of every grid point (x, y) in the interior of the physical domain
- Method developed by Thomson et al., 1985 (see ref)

Grid Generation for Structured Grids:
Differential Equation Methods, Example

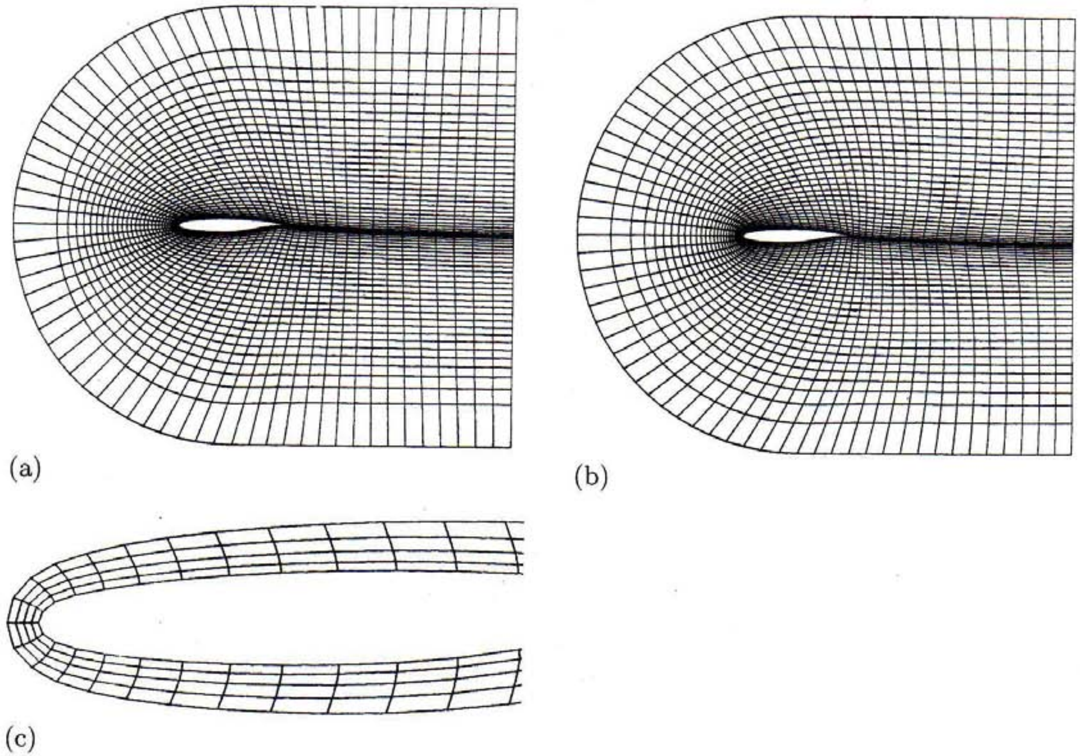


Fig. 9.13. **(a)** Starting algebraic C-grid around an airfoil section; 70 × 30 grid points; inner spacing $\Delta S_1 = 0.015c$, outer spacing $\Delta S_2 = 0.3c$, **(b)** Elliptic C-grid obtained after smoothing the algebraic grid of **(a)** by the solution of Poisson equations (50 iterations), **(c)** Close-up of the C-grid showing the application of orthogonality conditions near the leading edge region.

Grid Generation for Structured Grids:
Conformal Mapping Methods

- Conformal mapping schemes are analytical or partially analytical (as opposed to differential equation methods)
- Restricted to two dimensional flows (based on complex variables): useful for airfoils
- Examples:

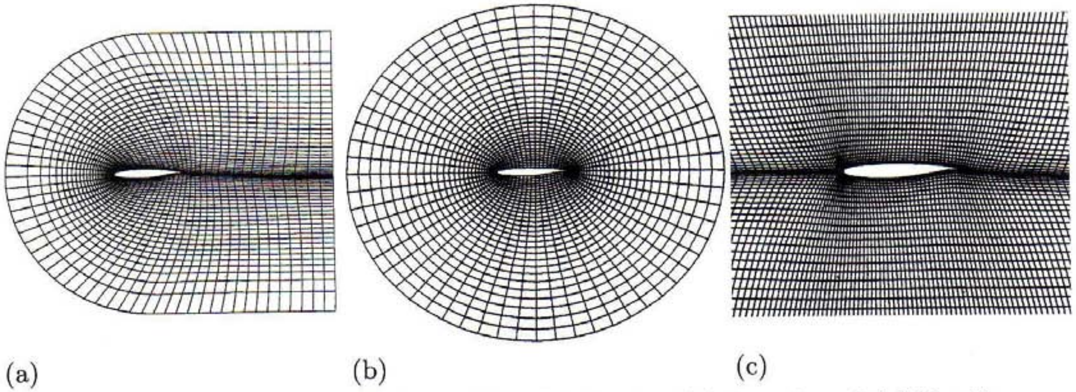


Fig. 9.14. Three common grids for airfoils. (**a**) C-grid, (**b**) O-grid, and (**c**) H-grid.

  – C-mesh: high density near leading edge of airfoil and good wake
  – O-mesh: high density near leading and trailing edge of airfoil
  – H-mesh: two sets of mesh lines similar to a Cartesian mesh, which is easiest to generate. Its mesh lines are often well aligned with streamlines

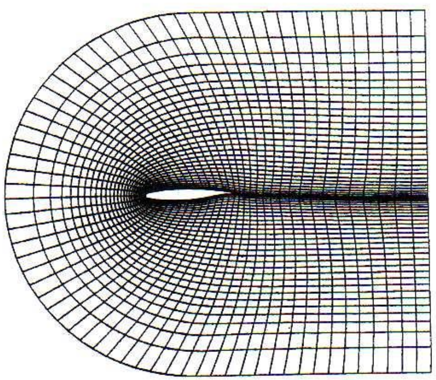
Grid Generation for Structured Grids:
Conformal Mapping Methods: Example

- C-mesh example is generated by a parabolic mapping function
- It is essentially a set of confocal, orthogonal parabolas wrapping around the airfoil
- The mapping is defined by:

$$2(x+iy)=(\xi+i\eta)^2$$

or

$$2\,x=\xi^2-\eta^2\,;\quad y=\xi$$



- Inverse transformation:

$$\xi^2=\sqrt{x^2+y^2}+x\,;\quad \eta^2=\sqrt{x^2+y^2}-x$$

- Polar coordinates can be used for easier "physical plane" to "computational plane" transformation.
- In conformal mapping, singular point is point where mapping fails (here, it is the origin) => move it to half the distance from the nose radius

Grid Generation: Unstructured Grids

- Generating unstructured grid is complicated but now relatively automated in "classic" cases
- Involves succession of smoothing techniques that attempt to align elements with boundaries of physical domain
- Decompose domain into blocks to de-couple the problems
- Need to define point positions and connections
- Most popular algorithms:
  - – Delaunay Triangulation Method
  - – Advancing Front Method
- Two schools of thought: structured vs. unstructured, what is best for CFD?
  - – Structured grids: simpler grid and straightforward treatment of algebraic system, but mesh generation constraints on complex geometries
  - – Unstructured grids: generated faster on complex domains, easier mesh refinements, but data storage and solution of algebraic system more complex

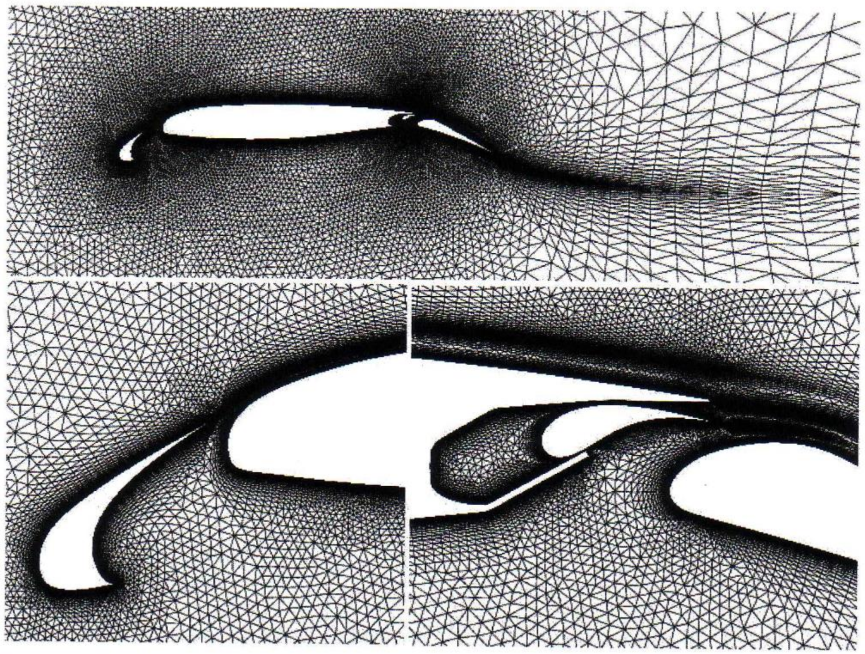


Fig. 9.16. 2D Unstructured grid for Navier–Stokes computations of a multi-element airfoil generated with the hybrid advancing front Delaunay method of Mavriplis [6].

Grid Generation: Unstructured Grids

- **Delaunay Triangulation (DT)**
  - Use a simple criterion to connect points to form conforming, non-intersecting elements
  - Maximizes minimum angle in each triangle
  - Not unique
  - Task of point generation is done independently of connection generation
- Based on Dirichlet's domain decomposition into a set of packed convex regions:
  - For a given set of points P, the space is subdivided into regions in such a way that each region is the space closer to P than to any other point = Dirichlet tessellation

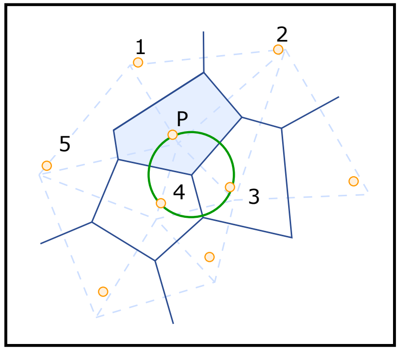


Image by MIT OpenCourseWare.

Note: at the end, points P are at summits of triangles

- This geometrical construction is known as the **Dirichlet (Voronoi) tessellation**
- The tessellation of a closed domain results in a set of non-overlapping convex regions called **Voronoi regions/polygons**
- The sides of the polygon around P is made of segments bisectors of lines joining P to its neighbors: if all pair of such P points with a common segment are joined by straight lines, the result is a Delaunay Triangulation
- Each vortex of a Voronoi diagram is then the circumcenter of the triangle formed by the three points of a Delaunay triangle
- Criterion: the circumcircle can not contain any other point than these three points

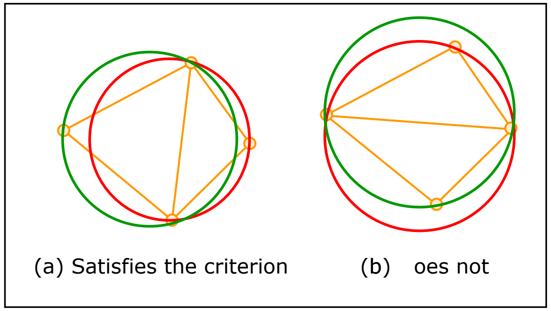


Image by MIT OpenCourseWare.

Grid Generation: Unstructured Grids

- **Advancing Front Method**
  - In this method, the tetrahedras are built progressively, inward from the boundary
  - An active front is maintained where new tetrahedra are formed
  - For each triangle on the edge of the front, an ideal location for a new third node is computed
  - Requires intersection checks to ensure triangles don't overlap

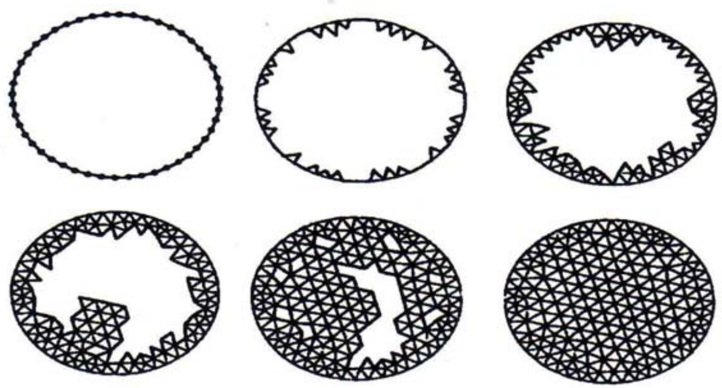


Fig. 9.20. Advancing Front technique for unstructured grid generation.

- In 3D, the Delaunay Triangulation is preferred (faster)

References and Reading Assignments Finite Element Methods

- Chapters 31 on "Finite Elements" of "Chapra and Canale, Numerical Methods for Engineers, 2006."
- Lapidus and Pinder, 1982: Numerical solutions of PDEs in Science and Engineering.
- Chapter 5 on "Weighted Residuals Methods" of Fletcher, Computational Techniques for Fluid Dynamics. Springer, 2003.

- Some Refs on Finite Elements only:
  - Hesthaven J.S. and T. Warburton. Nodal discontinuous Galerkin methods, vol. 54 of Texts in Applied Mathematics. Springer, New York, 2008. Algorithms, analysis, and applications
  - Mathematical aspects of discontinuous Galerkin methods (Di Pietro and Ern, 2012)
  - Theory and Practice of Finite Elements (Ern and Guermond, 2004)

FINITE ELEMENT METHODS: Introduction

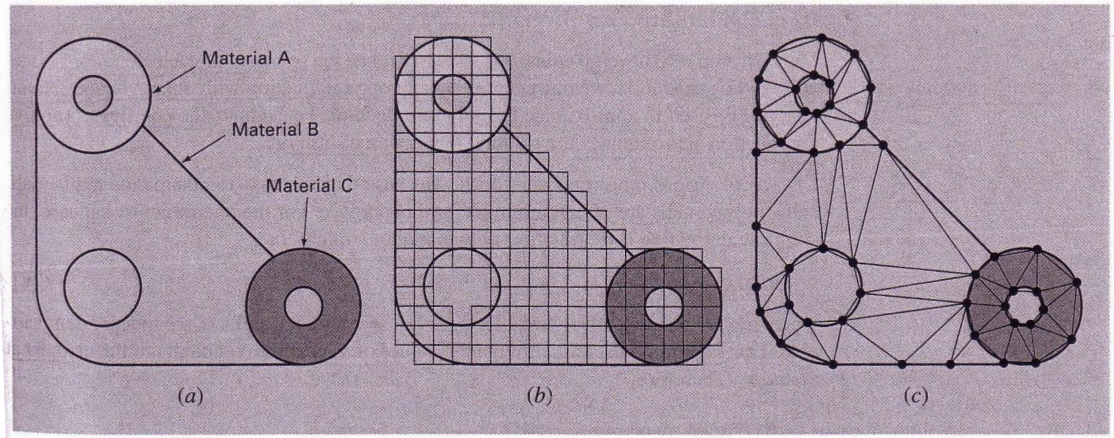
- Finite Difference Methods: based on a discretization of the differential form of the conservation equations
  - Solution domain divided in a grid of discrete points or nodes
  - PDE replaced by finite-divided differences = "point-wise" approximation
  - Harder to apply to complex geometries
- Finite Volume Methods: based on a discretization of the integral forms of the conservation equations:
  - Grid generation: divide domain into set of discrete control volumes (CVs)
  - Discretize integral equation
  - Solve the resultant discrete volume/flux equations
- Finite Element Methods: based on reformulation of PDEs into minimization problem, pre-assuming piecewise shape of solution over finite elements
  - Grid generation: divide the domain into simply shaped regions or "elements"
  - Develop approximate solution of the PDE for each of these elements
  - Link together or assemble these individual element solutions, ensuring some continuity at inter-element boundaries => PDE is satisfied in piecewise fashion

Finite Elements: Introduction, Cont'd

- Originally based on the Direct Stiffness Method (Navier in 1826) and Rayleigh-Ritz, and further developed in its current form in the 1950's (Turner and others)
- Can replace somewhat "ad-hoc" integrations of FV with more rigorous minimization principles
- Originally more difficulties with convection-dominated (fluid) problems, applied to solids with diffusion-dominated properties

Comparison of FD and FE grids

(a) A gasket with irregular geometry and nonhomogeneous composition. (b) Such a system is very difficult to model with a finite-difference approach. This is due to the fact that complicated approximations are required at the boundaries of the system and at the boundaries between regions of differing composition. (c) A finite-element discretization is much better suited for such systems.



Examples of Finite elements

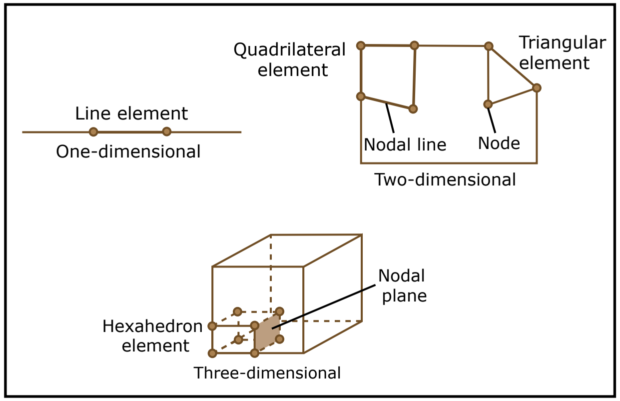


Image by MIT OpenCourseWare.

Finite Elements: Introduction, Cont'd

- Classic example: Rayleigh-Ritz / Calculus of variations
  - Finding the solution of $\dfrac{\partial^2 u}{\partial x^2} = -f \quad \text{on }]0,1[$

    is the same as finding u that minimizes $J(u) = \displaystyle\int_0^1 \frac{1}{2}\left(\frac{\partial u}{\partial x}\right)^2 - u\, f\, dx$

  - R-R approximation:
    - Expand unknown u into shape/trial functions $u(x) = \displaystyle\sum_{i=1}^{n} a_i\, \phi_i(x)$

      and find coefficients a_i such that $J(u)$ is minimized

- Finite Elements:
  - As Rayleigh-Ritz but choose trial functions to be piecewise shape function defined over set of elements, with some continuity across elements

Finite Elements: Introduction, Cont'd
Method of Weigthed Residuals

- There are several avenues that lead to the same FE formulation
  - A conceptually simple, yet mathematically rigorous, approach is the Method of Weighted Residuals (MWR)
  - Two special cases of MWR: the Galerkin and Collocation Methods
- In the MWR, the desired function u is replaced by a finite series approximation into shape/basis/interpolation functions:

$$\tilde{u}(x) = \sum_{i=1}^{n} a_i \, \phi_i(x)$$

  - $\phi_i(x)$ chosen such they satisfy the boundary conditions of the problem
  - But, they will not in general satisfy the PDE: $L(u) = f$

    $\Rightarrow$ they lead to a residual: $L(\tilde{u}(x)) - f(x) = R(x) \neq 0$
  - The objective is to select the undetermined coefficients a_i so that this residual is minimized in some sense

Finite Elements: Method of Weigthed Residuals, Cont'd

- One possible choice is to set the integral of the residual to be zero. This only leads to one equation for n unknowns

$\Rightarrow$ Introduce the so-called weighting functions $w_i\,(x)\; i=1,2,\ldots,\; n,$ and set the integral of each of the weighted residuals to zero to yield n independent equations:

$$\int_t \int_0^L R(x)\, w_i(x)\, dx\, dt = 0, \quad i = 1,2,\ldots,n$$

- In 3D, this becomes:

$$\int_t \int_V R(\mathbf{x})\, w_i(\mathbf{x})\, d\mathbf{x}\, dt = 0, \quad i = 1,2,\ldots,n$$

- A variety of FE schemes arise from the definition of the weighting functions and of the choice of the shape functions
  - Galerkin: the weighting functions are chosen to be the shape functions (the two functions are then often called basis functions or test functions)
  - Subdomain method: the weighting function is chosen to be unity in the sub-region over which it is applied
  - Collocation Method: the weighting function is chosen to be a Dirac-delta

Finite Elements: Method of Weigthed Residuals, Cont'd

- **Galerkin:** $\int_t \int_V R(\mathbf{x})\, \phi_i(\mathbf{x})\, d\mathbf{x}\, dt = 0, \quad i = 1, 2, \ldots, n$
  - Basis functions formally required to be complete set of functions
  - Can be seen as "residual forced to zero by being orthogonal to all basis functions"
- **Subdomain method:**

$$\int_t \int_{V_i} R(\mathbf{x})\, d\mathbf{x}\, dt = 0, \quad i = 1, 2, \ldots, n$$

  - Non-overlapping domains V_i often set to elements
  - Easy integration, but not as accurate

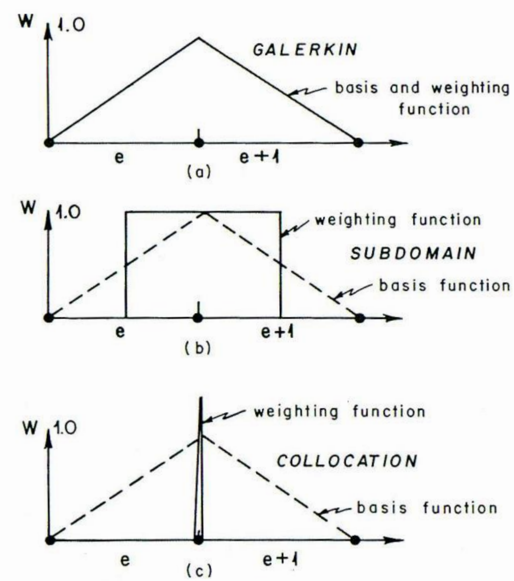


Figure 2.4. Schematic representation of the one-dimensional weighting functions for the Galerkin, subdomain and collocation methods. (It is assumed here that the chapeau function is used as a basis for all methods.)

- **Collocation Method:** $\int_t \int_V R(\mathbf{x})\, \delta_{x_i}(\mathbf{x})\, d\mathbf{x}\, dt = 0, \quad i = 1, 2, \ldots, n$
  - Mathematically equivalent to say that each residual vanishes at each collocation points $x_i \Rightarrow$ Accuracy strongly depends on locations x_i.
  - Requires no integration.